Homework 4: Blackjack

Course: CS 221 Spring 2019
Name: Bryan Yaggi

The search algorithms explored in the previous assignment work great when you know exactly the results of your actions. Unfortunately, the real world is not so predictable. One of the key aspects of an effective AI is the ability to reason in the face of uncertainty.

Markov decision processes (MDPs) can be used to formalize uncertain situations. In this homework, you will implement algorithms to find the optimal policy in these situations. You will then formalize a modified version of Blackjack as an MDP, and apply your algorithm to find the optimal policy.

Problem 1: Value Iteration

In this problem, you will perform the value iteration updates manually on a very basic game just to solidify your intuitions about solving MDPs. The set of possible states in this game is $\{-2, -1, 0, 1, 2\}$. You start at state 0, and if you reach either -2 or 2, the game ends. At each state, you can take one of two actions: $\{-1, +1\}$.

If you're in state s and choose -1:

- You have an 80% chance of reaching the state $s - 1$.
- You have a 20% chance of reaching the state $s + 1$.

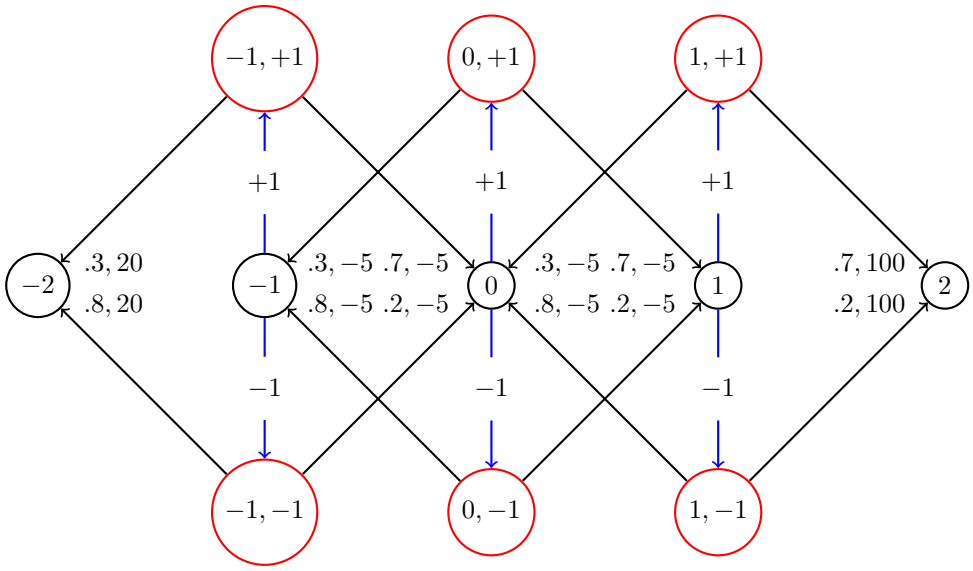
If you're in state s and choose $+1$:

- You have an 70% chance of reaching the state $s + 1$.
- You have a 30% chance of reaching the state $s - 1$.

If your action results in transitioning to state -2, then you receive a reward of 20. If your action results in transitioning to state 2, then your reward is 100. Otherwise, your reward is -5. Assume the discount factor γ is 1.

(a) Give the value of $V_{opt}(s)$ for each state s after 0, 1, and 2 iterations of value iteration. Iteration 0 just initializes all the values of V to 0. Terminal states do not have any optimal policies and take on a value of 0.

(b) What is the resulting optimal policy π_{opt} for all non-terminal states?



$$V_{opt}^{(t)}(s) = max_{a \in actions(s)} \sum_{s'} T(s,a,s')[reward(s,a,s') + \gamma V_{opt}^{(t-1)}(s')]$$

Iteration 0:

s	-2	-1	0	1	2
V_{opt}	0	0	0	0	0
π_{opt}	*none*	*none*	*none*	*none*	*none*

$$V_{opt}^{(1)}(-1) = max(.7(-5+1(0)) + .3(20+1(0)), .8(20+1(0)) + .2(-5+1(0))) = 15$$
$$V_{opt}^{(1)}(0) = max(.7(-5+1(0)) + .3(-5+1(0)), .8(-5+1(0)) + .2(-5+1(0))) = -5$$
$$V_{opt}^{(1)}(1) = max(.7(100+1(0)) + .3(-5+1(0)), .8(-5+1(0)) + .2(100+1(0))) = 68.5$$

Iteration 1:

s	-2	-1	0	1	2
V_{opt}	0	15.0	−5.0	68.5	0
π_{opt}	*none*	-1	*either*	$+1$	*none*

$$V_{opt}^{(2)}(-1) = max(.7(-5+1(-5)) + .3(20+1(0)), .8(20+1(0)) + .2(-5+1(-5))) = 14$$
$$V_{opt}^{(2)}(0) = max(.7(-5+1(68.5)) + .3(-5+1(15)), .8(-5+1(15)) + .2(-5+1(68.5))) = 47.45$$
$$V_{opt}^{(2)}(1) = max(.7(100+1(0)) + .3(-5+1(-5)), .8(-5+1(-5)) + .2(100+1(0))) = 67$$

Iteration 2:

s	-2	-1	0	1	2
V_{opt}	0	14.0	47.45	67.0	0
π_{opt}	*none*	$+1$	$+1$	$+1$	*none*